



Software Engineering

Lecture 1

Topics covered



✧ Software development

- What is meant by software engineering.

✧ Software engineering fundamentals

- A brief introduction that affect software engineering.

Software Engineering



- ✧ Software Engineering is the **process of designing, building and maintaining software systems**. It involves everything from understanding what the software needs to do, to planning and designing it, then developing it, and finally ensuring it continues to work well after it is in use.
- ✧ This involves resolving issues and making ongoing enhancements to **ensure the software remains effective** and efficient over time.

Importance of software engineering



- ✧ **Quality and Reliability:** Software engineering ensures the software works as expected and is free from errors.
- ✧ **Cost-Effectiveness:** Proper planning and design reduce the risk of expensive fixes and rework, saving time and money.
- ✧ **Scalability:** It helps in building software that can grow and adapt to future needs and increasing demand.
- ✧ **User Satisfaction:** Well-designed software meets user needs and provides a better experience.

Importance of software engineering



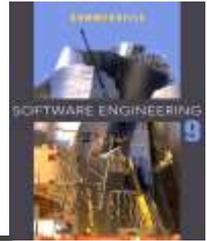
- ✧ **Maintenance:** It allows software to be easily updated and improved after its release, ensuring it stays relevant and functional over time.
- ✧ **Risk Management:** Software engineering identifies and addresses potential risks early in the development process.

Frequently asked questions about software engineering



Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

Frequently asked questions about software engineering



Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Software process activities



- ✧ **Software specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
- ✧ **Software development**, where the software is designed and programmed.
- ✧ **Software validation**, where the software is checked to ensure that it is what the customer requires.
- ✧ **Software evolution**, where the software is modified to reflect changing customer and market requirements.

Software engineering diversity



- ✧ There are many different types of software system and there is **no universal set of software techniques** that is applicable to all of these.
- ✧ The software engineering methods and tools used **depend on the type of application being developed**, the requirements of the customer and the background of the development team.

Application types



✧ Stand-alone applications

- These are application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.

✧ Interactive transaction-based applications

- Applications that execute on a remote computer and are accessed by users from their own PCs or terminals. These include web applications such as e-commerce applications.

✧ Embedded control systems

- These are software control systems that control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system.

Application types



✧ Batch processing systems

- These are business systems that are designed to process data in large batches. They process large numbers of individual inputs to create corresponding outputs.

✧ Entertainment systems

- These are systems that are primarily for personal use and which are intended to entertain the user.

✧ Systems for modeling and simulation

- These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.

Application types



✧ Data collection systems

- These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing.

✧ Systems of systems

- These are systems that are composed of a number of other software systems.

Software engineering fundamentals



- ✧ Some fundamental principles apply to all types of software system:
 - Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
 - Dependability and performance are important for all types of system.
 - Understanding and managing the software specification and requirements (what the software should do) are important.
 - Where appropriate, you should reuse software that has already been developed rather than write new software.

Software engineering and the web



- ✧ The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems.
- ✧ Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.
 - Users do not buy software buy pay according to use.

Web software engineering



- ✧ Software reuse is the dominant approach for constructing web-based systems.
 - When building these systems, you think about how you can assemble them from pre-existing software components and systems.
- ✧ Web-based systems should be developed and delivered incrementally.
 - It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.

Web-based software engineering



- ✧ Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.
- ✧ The fundamental ideas of software engineering, discussed in the previous section, apply to web-based software in the same way that they apply to other types of software system.



Thank You